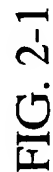


FIG. 1

FIG. 2



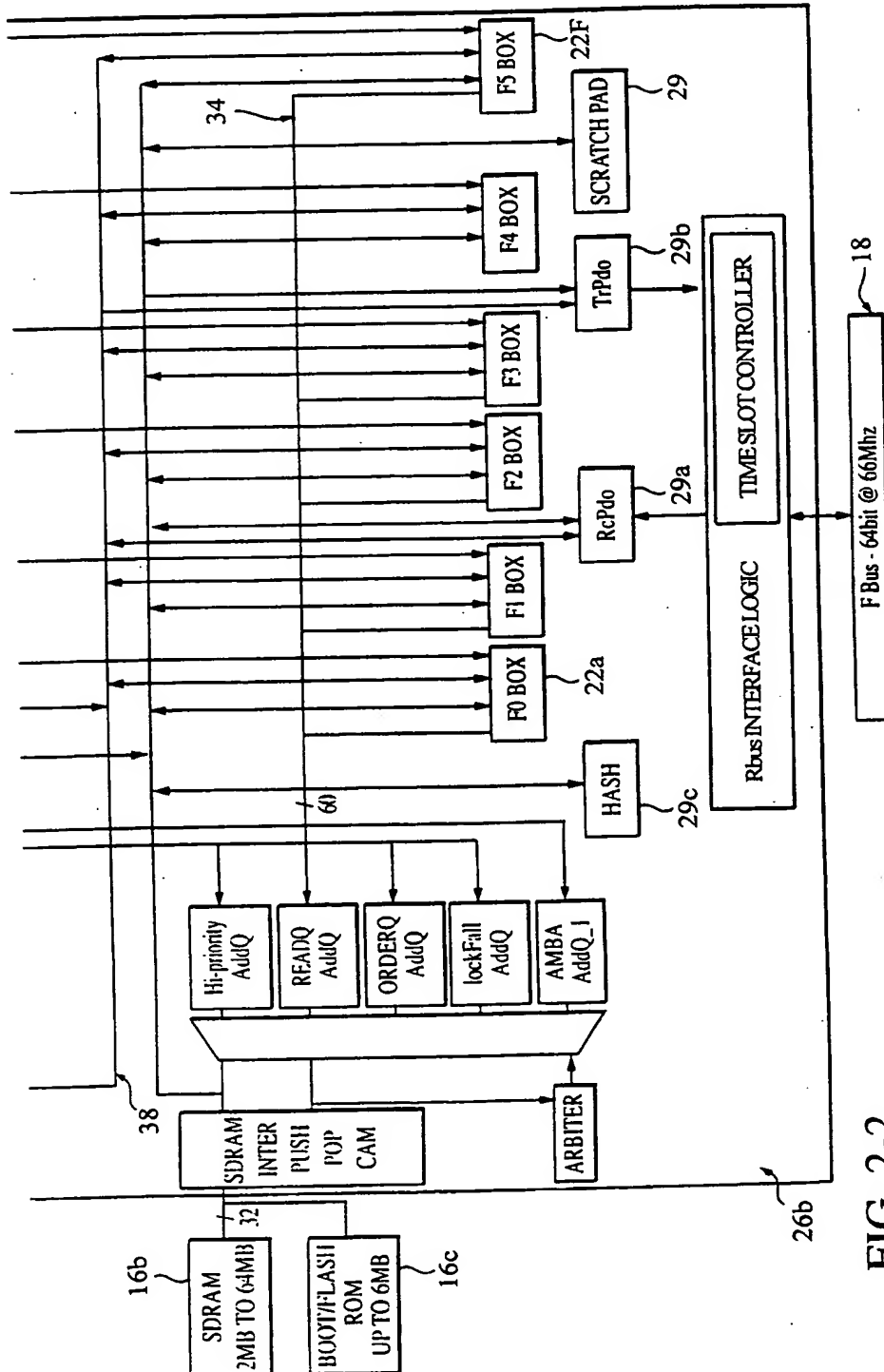


FIG. 2-2

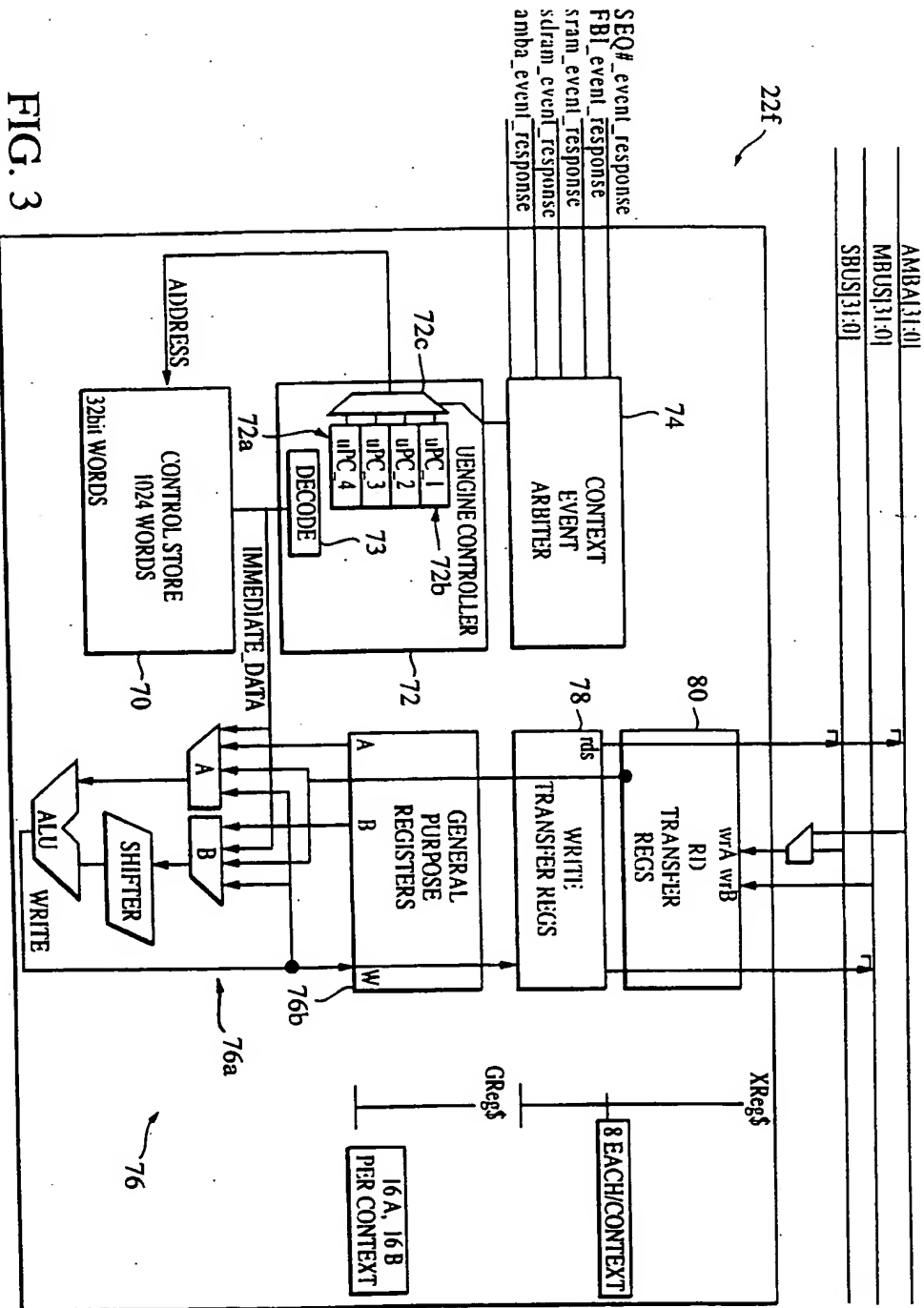


FIG. 3

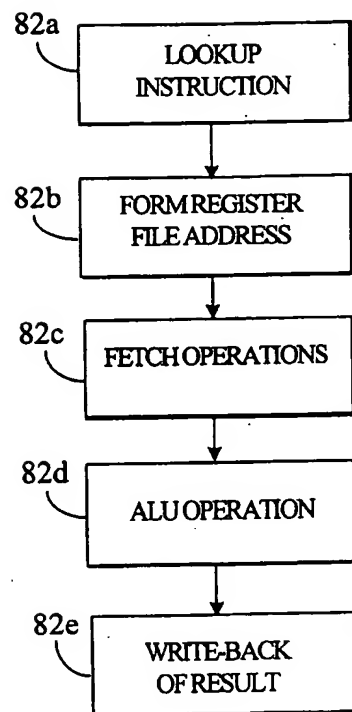


FIG. 4

ALU/SHIFT (set cc)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	sw	shift	rel	dest	reg			amount	rs	A	rel	source	B	rel	source	ro	im	Bi	ALUOp											
ALU/SHIFT (set cc)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	sw	shift	rel	dest	reg			amount		A	rel	source	B	rel	source				ALUOp											
ALU/SHIFT (set cc)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	sw	shift	rel	dest	reg			amount		A	rel	source	immediate							ALUOp										
ALU/SHIFT (set cc)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	1	0	0		dest	reg			sw	A	absolute	source	lo	B	Abs	Sec	Up	B	Srl	ALUOp												

Shift Decode:

(rs,r0) decode ([31:0] shifts into [63:32] and take [63:32]):

00 = left rotate

01 = right shift (32-ShfAmt = Right Shift Amt)

10 = left shift

11 = double shift (upper A-op shifts into lower B-op)

====> "left rotate" of zero gives zero shift (therwise zero amount signifies indirect shift)

ALU-OP decode:

0000 = B	0100 = ~A&B (~and)	1000 = A-B	1100 = A+B(8)
0001 = ~B	0101 = XOR	1001 = B-A	1101 = A+B(16)
0010 = A&B (and)	0110 = OR	1010 =	1110 = A+B
0011 = A&~B (and~)	0111 = mul-stuff	1011 =	1111 = A+B+Cin

FIG. 5